

The opinion in support of the decision being entered today was not written for publication and is not binding precedent of the Board.

Paper No. 22

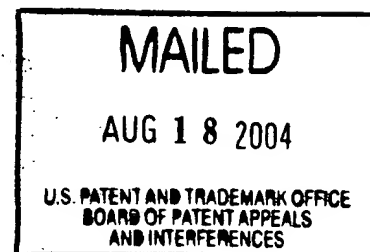
UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS  
AND INTERFERENCES

Ex parte MARION D. SKEEN

Appeal No. 2003-0319  
Application No. 09/206,663

ON BRIEF



Before JERRY SMITH, GROSS, and LEVY, Administrative Patent Judges.  
LEVY, Administrative Patent Judge.

DECISION ON APPEAL

This is a decision on appeal under 35 U.S.C. § 134 from the examiner's final rejection of claims 1-20, which are all of the claims pending in this application.

BACKGROUND

Appellant's invention relates to a real-time decision support system. An understanding of the invention can be derived

from a reading of exemplary claim 1, which is reproduced as follows:

1. A computer-implemented real-time decision support system comprising:

means for receiving complex queries posed by a plurality of users;

means for providing for each of the queries an executable query process furnishing a result when data sources satisfy constraints of the query;

means for providing data sources, the data sources including an event source;

means for responding to an event furnished by an event source to initiate execution of all query processes having constraints satisfied by data sources; and

means for furnishing results of executed query processes to users.

The prior art reference of record relied upon by the examiner in rejecting the appealed claims is:

Risberg et al. (Risberg)	5,339,392	Aug. 16, 1994
--------------------------	-----------	---------------

Claims 1-20 stand rejected under 35 U.S.C. § 102(b) as being anticipated by Risberg.

Rather than reiterate the conflicting viewpoints advanced by the examiner and appellant regarding the above-noted rejection, we make reference to the examiner's answer (Paper No. 16, mailed August 14, 2002) for the examiner's complete reasoning in support

of the rejection, and to appellant's brief (Paper No. 15, filed June 28, 2002) for appellant's arguments thereagainst. Only those arguments actually made by appellant have been considered in this decision. Arguments which appellant could have made but chose not to make in the brief have not been considered. See 37 CFR 1.192(a).

#### OPINION

In reaching our decision in this appeal, we have carefully considered the subject matter on appeal, the rejection advanced by the examiner, and the evidence of anticipation relied upon by the examiner as support for the rejection. We have, likewise, reviewed and taken into consideration, in reaching our decision, appellant's arguments set forth in the brief along with the examiner's rationale in support of the rejection and arguments in rebuttal set forth in the examiner's answer.

Upon consideration of the record before us, we reverse, essentially for the reasons set forth by appellant. We begin with claim 1.

To anticipate a claim, a prior art reference must disclose every limitation of the claimed invention, either explicitly or

inherently. In re Schreiber, 128 F.3d 1473, 1477, 44 USPQ2d 1429, 1431 (Fed. Cir. 1997). As stated in In re Oelrich, 666 F.2d 578, 581, 212 USPQ 323, 326 (CCPA 1981) (quoting Hansgird v. Kemmer, 102 F.2d 212, 214, 40 USPQ 665, 667 (CCPA 1939))

(internal citations omitted):

Inherency, however, may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient. If, however, the disclosure is sufficient to show that the natural result flowing from the operation as taught would result in the performance of the questioned function, it seems to be well settled that the disclosure should be regarded as sufficient.

Appellant asserts (brief, page 4) that Risberg does not disclose the function of responding to an event to execute processes for all queries having constraints satisfied by a data source. Appellant further asserts (id.) that Risberg discloses scripted commands executed in sequence, not concurrently. Appellant acknowledges (id.) that Risberg discloses an event driven publish/subscribe architecture similar to the claimed invention, and notes that in Risberg, scripted commands can be executed if the data field exceeds a limit. Appellant argues that as set forth in independent claims 1 and 11, the claimed invention receives complex queries which are declarative and not

programmatic (brief, page 4). Pointing to page 13 of the specification, appellant asserts that appellant's queries are structured in Object Query Language (OQL) which includes declarative statements, but lacks executable commands (brief, page 5). Appellant argues that Risberg discloses scripts which are merely display instructions or other executable commands (brief, page 5). It is further argued that Risberg's scripts are executable commands, and these scripts are not queries (brief, page 5-6). Appellant concurs with the examiner that Risberg could utilize queries to select desired data, but argues (brief, page 6) that Risberg fails to disclose responding to an event by initiating execution of all query processes having constraints satisfied by the data source. Appellant further disagrees with examiner's interpretation of Risberg that an object can be a query and that a script is an object, and therefore - a script can be a query (brief, page 6).

The examiner, relying on Fig. 4 and 5, equates Risberg's reference to queries to the claimed complex queries. The examiner maintains that Risberg discloses queries responding to events which execute scripts containing constraints based on data sources (answer, page 4). Additionally, the examiner asserts that Risberg discloses constraint indexing and related comparison

to yield a set of queries having constraints satisfied by an event wherein updates from a database are dispatched such as a stock quote (answer, page 4).

A review of Risberg reveals that the reference relates to a program application whereby a user, using a collection of layout tools, can create an application to be displayed.

From our review of Risberg, we find that Risberg discloses a program whereby a user, using a collection of layout tools may define an active document (Risberg, col. 2, lines 27-31). The layout, which will appear on all sheets, will include one to four margins, headers or footers with text, a common button or a set of buttons that will display objects with user programmable text labels, actuated by push buttons (col. 2, lines 15-22). A button tool can be used to create display objects that will execute scripted actions when the button is pushed. The scripted actions are entered by the user in whatever sequence desired (col. 4, lines 4-7).

From the teachings of Risberg, we find that although Risberg refers to queries and scripts, that in Risberg, the application

refers to simple queries and to scripts that are executable commands<sup>1</sup> (col. 4, lines 4-8).

From our review of the record, we find that a "complex query" is a term of art, and we agree with appellant that an artisan working in the Object Query Language would consider a complex query to be a declarative specification, and not an executable command. We take Notice that a script is not a complex query<sup>2</sup>. From the examiner's assertion (answer, page 4) that "Examiner maintains that Risberg discloses such wherein scripts are user defined commands such as macros which are well known to be stored persistently as they can be executed by pressing a button or icon and can be copied" we find that the scripts of Risberg, which are executable commands, are not

---

1

IBM Dictionary of Computing, ©1994, query: (4) A request for information from a file based on specific conditions; for example, a request for a list of all customers whose balance is greater than \$1000.

IBM Technical Disclosure Bulletin NN921158, November 1992. "The user is able to achieve a very complex query by iteratively combining and refining simple queries. . . . These combination objects can be further combined with other combination objects or with simple queries to form more complex queries."

<sup>2</sup> Microsoft Computer Dictionary, ©1994, script: A type of program that consists of a set of instructions to an application or utility program. A script usually consists of instructions expressed using the application's or utility's rules and syntax, combined with simple control structures such as loops and if/then expressions.

complex queries as set forth in independent claim 1, as the phrase would be understood by an artisan. Accordingly, we agree with appellant (brief, pages 4 and 6) that Risberg fails to disclose "receiving complex queries." In addition, because the scripts of Risberg are executed sequentially, we find that Risberg does not disclose "means for responding to an event to initiate execution of all query processes having constraints satisfied by data sources" since Risberg does not respond to an event by initiating execution of all query processes having constraints satisfied by data sources, as recited in claim 1.

From all of the above, we find that the examiner has failed to establish a prima facie case of anticipation of claim 1. The rejection of claim 1 and claims 2-10, dependent therefrom, is therefore reversed. In addition, as independent claim 11 similarly recites "receiving complex queries" and "responding to events by executing all query processes having constraints satisfied by data sources to provide results required by each of the queries," we find that Risberg does not anticipate independent claim 11. Accordingly, the rejection of claim 11, and claims 12-20, dependent therefrom, under 35 U.S.C. § 102(b) is reversed.



To summarize, the decision of the examiner to reject claims 1-20 under 35 U.S.C. § 102(b) is reversed.

*Jerry Smith*  
JERRY SMITH

Anita Pellman Grosse

BOARD OF PATENT  
APPEALS  
AND  
INTERFERENCES

STUART S. LEVY

SSL/kis

Appeal No. 2003-0319  
Application No. 09/206,663

Page 10

MARK S. KAUFMAN  
NIXON & PEABODY, LLP  
8180 GREENSBORO DRIVE  
SUITE 800  
MCLEAN, VA 22102

# IBM DICTIONARY OF COMPUTING

*Compiled and edited by*  
**GEORGE McDANIEL**

11-08-00A09:42 RCVD

**McGRAW-HILL, INC.**  
New York San Francisco Washington, D.C. Auckland Bogotá  
Caracas Lisbon London Madrid Mexico City Milan  
Montreal New Delhi San Juan Singapore  
Sydney Tokyo Toronto

### **Limitation of Liability**

While the Editor and Publisher of this book have made reasonable efforts to ensure the accuracy and timeliness of the information contained herein, neither the Editor nor the Publisher shall have any liability with respect to loss or damage caused or alleged to be caused by reliance on any information contained herein.

Copyright © 1994 by International Business Machines Corporation. All rights reserved. Printed in the United States of America. Except as permitted under the United States Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a data base or retrieval system, without the prior written permission of the publisher.

7 8 9 0 DOC/DOC 9 9 8

ISBN 0-07-031488-8 (HC)  
ISBN 0-07-031489-6 (PBK)

*The sponsoring editor for this book was Daniel A. Gonneau and the production supervisor was Thomas G. Kowalczyk.*

*Printed and bound by R. R. Donnelley & Sons Company.*

### **Tenth Edition (August 1993)**

This is a major revision of the *IBM Dictionary of Computing*, SC20-1699-8, which is made obsolete by this edition. Changes are made periodically to the information provided herein.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country. Comments may be addressed to IBM Corporation, Department E37/656, P. O. Box 12195, Research Triangle Park, NC 27709.

### **International Edition**

Copyright © 1994 by International Business Machines Corporation. Exclusive rights by McGraw-Hill, Inc. for manufacture and export. This book cannot be re-exported from the country to which it is consigned by McGraw-Hill. The International Edition is not available in North America.

When ordering this title, use ISBN 0-07-113383-6.

## qualified object name

or minimized is a quadratic constraints are linear. Contrast with convex programming, integer programming, mathematical programming, nonlinear

state of being at 90 degrees TSC video, the state of a ng 90 degrees out of phase. See also in-phase (I).

Four registers that function  
A) Synonymous with qua-

onym for quadruple-length

processing, a function that up of characters at any point .)

ualification.

all that contains at least one

COBOL, an identifier that is ie, followed by one or more nectives OF and IN followed

ob name, a related user name, number. Contrast with job

control (QLLC) An X.25 pro- sfer of data link control infor- djoining SNA nodes that are X.25 packet-switching data ovides the qualifier "Q" bit in ntify packets that carry logical on. Contrast with enhanced .LC), physical services header

data name explicitly accompa- of the class to which it belongs tion system. (I) (A) (2) A ade unique by the addition of (3) In the AS/400 system, the taining an object and the name with object name.

An object name and the name ing the object. Contrast with

## qualified segment search argument (SSA)

[549]

## query command

**qualified segment search argument (SSA)** In IMS/VS, a segments search argument (SSA) that contains, in addition to the segment name, one or more qualification statements. A qualified SSA describes the segment type and occurrence that is to be accessed.

**qualifier** (1) A modifier that makes a name unique. (2) All names in a qualified name other than the right-most, which is called the simple name. (3) In COBOL, a data-name or a name associated with a level indicator that is used in a reference either together with another data-name which is the name of an item that is subordinate to the qualifier or together with a condition-name. (4) In COBOL, a section-name that is used in a reference together with a paragraph-name specified in that section. (5) In COBOL, a library-name that is used in a reference together with a text-name associated with that library.

**qualifier bit** See Q-bit.

**quality assurance (QA)** (1) The planned systematic activities necessary to ensure that a component or system conforms to established technical requirements. (T) (2) All actions that are taken to ensure that a development or organization delivers products that meet performance requirements and adhere to standards and procedures. (A) (3) The in an enterprise for the purpose of providing and maintaining some degree of confidence in data integrity and accuracy throughout the life cycle of the data, which includes input, update, manipulation, and output. (A)

**quality of service (QOS)** A set of communication characteristics required by an application. Each QOS defines a specific transmission priority, level of route reliability, and security level. Each QOS also defines whether the sessions are interactive. Contrast with class of service (COS).

**quantization** The subdivision of the range of values of a variable into a finite number of nonoverlapping, but not necessarily equal, subranges or intervals, each of which is represented by an assigned value within the subrange; for example, a person's age is quantized for most purposes with a quantum of one year. (A)

**quantize** To divide the range of a variable into a finite number of nonoverlapping intervals that are not necessarily of equal width, and to designate each interval by an assigned value within the interval; for example, a person's age is for many purposes quantized with a quantum (interval) of one year. (I) (A)

**quantizing** The second step in the three-step process of converting an analog signal to a digital signal. The

three steps are: sampling, quantizing, and encoding. In quantizing, a sample is converted from an analog value to a digital value having a limited number of bits, which is then ready for encoding. See also encoding, sampling.

**quantizing noise** The artifact caused by quantizing with too few levels.

**quantum** A subrange in quantization. (A)

**quark** In AIX Enhanced X-Windows, synonym for string.

**quarter-speed** One-fourth the rated speed of the associated equipment; in transoceanic telegraph, one-fourth of full speed, or 12.5 baud or 16+ wpm.

**quarter-squares multiplier** An analog multiplier whose operation is based on the identity:

$$xy = [(x + y)^2 - (x - y)^2] / 4$$

incorporating inverters, analog adders, and square-law function generators. (I)

**quartet** A byte composed of four binary elements. (I) (A) Synonymous with four-bit byte.

**quartz iodide light** In video or film production, a very bright light source consisting of a thick glass or quartz bulb filled with a halogen gas, such as iodide, and containing an electrical filament that reaches a very high temperature during operation.

**quasistable state** Synonym for unstable state.

**quaternary operator** An operator that requires exactly four operands. (I) (A)

**query** (1) A request for data from a database, based on specified conditions; for example, a request for availability of a seat on a flight reservation system. (T) (2) The process by which a master station asks a slave station to identify itself and to give its status. (T) (3) In interactive systems, an operation at a terminal that elicits a response from the system. (4) A request for information from a file based on specific conditions; for example, a request for a list of all customers whose balance is greater than \$1000. (5) In System/38, a utility that is part of the Interactive Data Base Utilities licensed program. (6) In the AS/400 system, the query management object that is used to define queries against relational data.

**query application** See application.

**query command** In the AS/400 system, the name of an action, and any associated parameters, that can be

MICROSOFT PRESS®

# COMPUTER DICTIONARY

SECOND

EDITION



SCIENTIFIC  
INFORMATION CENTER  
OCT 27 1993

PATENT & TRADEMARK OFFICE

THE COMPREHENSIVE  
STANDARD FOR  
BUSINESS, SCHOOL,  
LIBRARY, AND HOME



LANCE LEONARD BARRY, ESQ.  
PRIMARY EXAMINER  
GROUP 2300

030

Barry, Lance

PUBLISHED BY  
Microsoft Press  
A Division of Microsoft Corporation  
One Microsoft Way  
Redmond, Washington 98052-6399

Copyright © 1994 by Microsoft Press

All rights reserved. No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher.

Library of Congress Cataloging-in-Publication Data

Microsoft Press computer dictionary : the comprehensive standard for  
business, school, library, and home / Microsoft Press. -- 2nd ed.  
p. cm.

ISBN 1-55615-597-2

1. Computers--Dictionaries. 2. Microcomputers--Dictionaries.

I. Microsoft Press. II. Title: Computer dictionary.

QA76.15.M54 1993

004'.03--dc20

93-29868  
CIP

Printed and bound in the United States of America.

1 2 3 4 5 6 7 8 9 MLML 9 8 7 6 5 4

Distributed to the book trade in Canada by Macmillan of Canada, a division of Canada Publishing Corporation.

Distributed to the book trade outside the United States and Canada by  
Penguin Books Ltd.

Penguin Books Ltd., Harmondsworth, Middlesex, England

Penguin Books Australia Ltd., Ringwood, Victoria, Australia

Penguin Books N.Z. Ltd., 182-190 Wairau Road, Auckland 10, New Zealand

British Cataloging-in-Publication Data available.

**Project Editor:** Casey D. Doyle

**Manuscript Editor:** Alice Copp Smith

**Technical Editors:** Mary DeJong, Jeff Carey, Dail Magee, Jr., Jim Fuchs, Seth McEvoy

QA

76.15

M54

1993

c.17



screen fonts are bit maps (pixel-by-pixel designs) that are used for the display and for printing on an ImageWriter or a LaserWriter printer. Often a screen font has a corresponding PostScript font for printing to a PostScript-compatible printer. *Compare* PostScript font; *see also* derived font, intrinsic font.

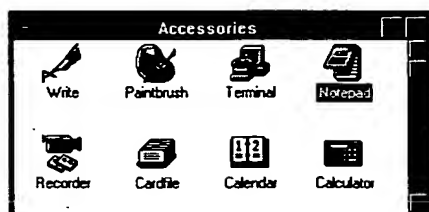
**screen frequency** *See* halftone.

**screen grabber** *See* grabber.

**script** A type of program that consists of a set of instructions to an application or utility program. A script usually consists of instructions expressed using the application's or utility's rules and syntax, combined with simple control structures such as loops and if/then expressions. *See also* macro.

**scroll arrow** *See* scroll bar.

**scroll bar** In many types of graphical user interface, a vertical or horizontal bar at the side or bottom of a window that can be used with a mouse for moving around in a document. *See* the illustration. A scroll bar has three active areas; a vertical scroll bar in a word-processing application, for example, has two scroll arrows for moving up and down by one line and a scroll box for moving to an arbitrary location in the document. Clicking in one or the other of the gray areas in the scroll bar moves up or down through the document in increments of one window.



Scroll bar

**Scroll bar.**

**scroll box** *See* scroll bar.

**scrolling** The process of moving a document in a window to permit viewing of any desired portion. Scrolling is so named because it is the electronic equivalent of reading through a rolled (scrolled) document rather than flipping through the pages of a book. The keyboard provides a number of "scroll-control" keys, such as the up,

down, left, and right arrow keys. Application programs often offer additional methods of scrolling—for example, combining the Control and Page Up keys to move to the beginning of a document. *See also* scroll bar.

**Scroll Lock key** On the IBM PC/XT and original AT and compatible keyboards, a key on the top row of the numeric keypad that governs the effect of the cursor-control keys. On the enhanced keyboard, this key is to the right of the function keys in the top row. When toggled on, the Scroll Lock key affects the cursor-control keys on the numeric keypad or, with an enhanced keyboard, the cursor-control keys between the numeric keypad and the main keyboard. The operation of the cursor keys with the Scroll Lock key toggled on depends on the application program; a word-processing program, for example, might scroll a document around the cursor when the cursor keys are used and Scroll Lock is on, whereas the cursor keys would move the cursor around the document if Scroll Lock is off. Although the name of this key suggests that pressing it stops the screen from scrolling, it does not do this in all situations. Instead, Pause (on the enhanced keyboard only), Ctrl-Num Lock, and Ctrl-S are the keys that stop scrolling of a DOS screen.

**SCSI** Pronounced "scuzzy"; acronym for small computer system interface, a standard high-speed parallel interface defined by the X3T9.2 committee of the American National Standards Institute (ANSI). A SCSI interface is used for connecting microcomputers to peripheral devices, such as hard disks and printers, and to other computers and local area networks. Up to seven devices, not including the computer, can be attached through a single SCSI connection (port) through sequential connections called a daisy chain. Each device has an address (priority number). Only one device at a time can transmit through the port; priority is given to the device with the highest address. A SCSI port is standard on some computers and can be installed in other computers by means of an expansion board. *See also* enhanced small device interface, IDE.

**scuzzy** *See* SCSI.



First HitPrevious DocNext DocGo to Doc#

Generate Collection

Print

L5: Entry 6 of 12

File: TDBD

Nov 1, 1992

TDB-ACC-NO: NN921158

DISCLOSURE TITLE: User Interface for Simplifying Complex Query Expressions.

## PUBLICATION-DATA:

IBM Technical Disclosure Bulletin, November 1992, US

VOLUME NUMBER: 35

ISSUE NUMBER: 6

PAGE NUMBER: 58 - 59

PUBLICATION-DATE: November 1, 1992 (19921101)

CROSS REFERENCE: 0018-8689-35-6-58

## DISCLOSURE TEXT:

- This article describes a graphical user-interface technique for constructing simple queries and using set operations to create more powerful, complex ones. The user is able to achieve a very complex query by iteratively combining and refining simple queries. - Many of today's computer applications make use of searching, e.g., looking for documents in an electronic library, looking for products in an electronic inventory, or looking for people in an electronic directory. These applications tend to make use of database technology. Databases are capable of handling very complex queries. However, queries tend to be limited by the ability of users to compose search expressions. Thus, most applications designers have simplified the interface used to express search criteria so that it is easy for users to define a search and the odds of a successful search execution are increased. These query interfaces tend to be fill in the blank interfaces. Users can easily find people who work for a specific manager or customers who spent so much money in January. - Unfortunately, this approach does not always let users get to the information they want. For example, how do users find the people who work for John or Sally or how do users find who ordered more than \$1000 but less than \$10,000 worth of supplies each January during the last five years? Using the proposed interface, application designers can still use fill in the blank techniques to allow users to easily and simply express search criteria. Users can save these simple queries and have access to them (the definitions and the results) as objects. Users can easily change search definitions and regenerate the results. - Users can logically combine (and, or, and difference operations) to get new results. Users can select multiple definitions, and the logical operation that is in effect is performed. The results of the operation are displayed. Thus, users can see the effect of operations as they occur. These operations may occur more quickly because they are performed with results that may have already been stored locally. - Users can also save a logical combination as an object. These combination objects can be further combined with other combination objects or with simple queries to form more complex queries. - Let us consider an example. A manager has built a query to show the successful products in region 3 and the successful products in region 4. (These queries could be complex queries that look at a number of different parameters to determine success.) The manager wants to know what the common top selling products are in both regions (intersection). This disclosure would allow the manager to select the two simpler query icons and create a new,

more complex one very easily without having to express the criteria in one complex interface. - These techniques would provide a graphical user interface technique for constructing simple queries and using set operations to create more powerful, complex ones. Thus the user would be able to achieve a very complex query by iteratively combining and defining simple queries.

SECURITY: Use, copying and distribution of this data is subject to the restrictions in the Agreement For IBM TDB Database and Related Computer Databases. Unpublished - all rights reserved under the Copyright Laws of the United States. Contains confidential commercial information of IBM exempt from FOIA disclosure per 5 U.S.C. 552(b)(4) and protected under the Trade Secrets Act, 18 U.S.C. 1905.

COPYRIGHT STATEMENT: The text of this article is Copyrighted (c) IBM Corporation 1992. All rights reserved.

[Previous Doc](#)

[Next Doc](#)

[Go to Doc#](#)